

LISTING OF CLAIMS

1. (currently amended) A method of operating a computer system including a virtual machine supporting an object-oriented environment, in which programs to run on the virtual machine are formed from classes loaded into the virtual machine by a class loader, wherein a class must be initialized before being used by a program, said method comprising the steps of:

starting a first application on a virtual machine, including loading and initializing a set of one or more classes for said first application;

running said first application;

after the first application has finished running, resetting at least one class from said set of one or more classes from the first application; and

starting a second application on the virtual machine, said second application initializing said at least one class that has been reset from the first application without having to load said at least one class into said virtual machine, prior to use of said at least one class by the second application.

2. (original) The method of claim 1, wherein each class has an initialization flag which is set when the class is initialized, and said step of resetting at least one class comprises resetting said initialization flag to its state prior to initialization.

3. (original) The method of claim 2, wherein said step of resetting further comprises setting variables to their default state prior to initialization.

4. (original) The method of claim 1, wherein the first and second applications run in a middleware environment, and a middleware program initiates said steps of starting a first application and starting a second application.

5. (original) The method of claim 4, wherein said step of loading the set of one or more classes for the first application is performed by an application class loader instance, and said middleware program initiates said step of resetting at least one class by calling a method of the application class loader.

6. (original) The method of claim 5, wherein the step of calling a method of the application class loader comprises passing a reference to said application class loader instance.

7. (original) The method of claim 6, wherein said step of resetting further comprises checking that said reference to said application class loader instance is the only valid remaining reference to said application class loader instance and the set of one or more classes loaded by it.

8. (original) The method of claim 5, wherein said method is a static method.

9. (original) The method of claim 5, further comprising the steps of the virtual machine maintaining a pool of application class loader instances and associated classes, and responding to said method call by returning to the middleware a reference to one of the application class loader instances from said pool.

10. (currently amended) A computer system including a virtual machines supporting an object-oriented environment,

in which programs to run on the virtual machine are formed from classes loaded into the virtual machine by a class loader, wherein a class must be initialized before being used by a program, said system comprising:

means for starting a first application on the virtual machine, including loading and initializing a set of one or more classes for said first application;

means for running said first application;

means responsive to the first application finishing running, for resetting at least one class from said set of one or more classes from the first application; and

means for starting a second application on the virtual machine, said second application initializing said at least one class that has been reset from the first application without having to load said at least one class into said virtual machine, prior to use of said at least one class by the second application.

11. (original) The system of claim 10, wherein each class has an initialization flag which is set when the class is initialized, and said means for resetting at least one class comprises means for resetting said initialization flag to its state prior to initialization.

12. (original) The system of claim 11, wherein said means for resetting further comprises means for setting variables to their default state prior to initialization.

13. (original) The system of claim 10, wherein the first and second applications run in a middleware environment, and a middleware program activates said means for starting a first application and starting a second application.

14. (original) The system of claim 13, wherein said means for loading the set of one or more classes for the first application comprises an application class loader instance, and said middleware program activates said means for resetting at least one class by calling a method of the application class loader.

15. (original) The system of claim 14, wherein calling a method of the application class loader involves passing a reference to said application class loader instance.

16. (original) The system of claim 15, wherein said means for resetting further comprises means for checking that said

reference to said application class loader instance is the only valid remaining reference to said application class loader instance and the set of one or more classes loaded by it.

17. (original) The system of claim 14, wherein said method is a static method.

18. (original) The system of claim 14, wherein the virtual machine includes a pool of application class loader instances and associated classes, and responds to said method call by returning to the middleware a reference to one of the application class loader instances from said pool.

19. (currently amended) A computer program product comprising a set of computer readable instructions recorded on a medium, said instructions implementing a virtual machine supporting an object-oriented environment, in which programs to run on the virtual machine are formed from classes loaded into the virtual machine by a class loader, wherein a class must be initialized before being used by a program, said virtual machine comprising:

means for starting a first application on the virtual machine, including loading and initializing a set of one or more classes for said first application;

means for running said first application;

means for resetting at least one class from said set of one or more classes from the first application after the first application has finished running; and

means for starting a second application, said second application causing the virtual machine to initialize said at least one class that has been reset from the first application without having to load the at least one class into said virtual machine, prior to use of said at least one class by the second application.

20. (original) The program product of claim 19, wherein each class has an initialization flag which is set when the class is initialized, and said means for resetting at least one class comprises means for resetting said initialization flag to its state prior to initialization.

21. (original) The program product of claim 20, wherein said means for resetting further comprises means for setting variables to their default state prior to initialization.

22. (original) The program product of claim 19, wherein the first and second applications run in a middleware environment, and a middleware program activates said means for starting a first application and starting a second application.

23. (currently amended) The program product of claim 22, wherein said ~~of~~ means for loading the set of one or more classes for the first application comprises an application class loader instance, and said middleware program activates said means for resetting at least one class by calling a method of the application class loader.

24. (original) The program product of claim 23, wherein calling a method of the application class loader involves passing a reference to said application class loader instance.

25. (original) The program product of claim 24, wherein said means for resetting further comprises means for checking that said reference to said application class loader instance is the only valid remaining reference to

said application class loader instance and the set of one or more classes loaded by it.

26. (original) The program product of claim 23, wherein said method is a static method.

27. (original) The program product of claim 23, wherein the virtual machine includes a pool of application class loader instances and associated classes, and responds to said method call by returning to the middleware a reference to one of the application class loader instances from said pool.